IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NETWORK APPLIANCE INC,

Plaintiff,

v.

SUN MICROSYSTEMS INC,

Defendant.
_____/

No. C-07-06053 EDL

ORDER GRANTING SUN MICROSYSTEMS INC.'S MOTION NO. 2 FOR SUMMARY JUDGMENT OF NON-INFRINGEMENT OF U.S. PATENT NO. 6,892,211

I. INTRODUCTION

On September 5, 2007, Network Appliance, Inc. ("NetApp") filed its Complaint, alleging that Sun Microsystems, Inc. ("Sun") infringed and is infringing, directly and indirectly under 35 U.S.C. § 271, certain of its patents, by making, using, selling, or offering for sale certain data processing systems and related software. NetApp seeks a declaratory judgment that certain patents owned by Sun are each not infringed, are invalid and/or are unenforceable, as well as a permanent injunction and damages. On October 25, 2007, Sun filed an Answer and Counterclaim, denying the material allegations of NetApp's Complaint and asserting a number of affirmative defenses and counterclaims. Sun denies infringing any of the NetApp Patents, including the patent at issue in this motion (U.S. Patent Number 6,892,211 (the "'211 patent")) and alleges that NetApp infringes a number of its patents instead. On September 10, 2008, this Court issued an Order Construing Claims (the "9/10/08 Order") in which it construed fourteen disputed terms and/or phrases contained in various claims in the seven patents at issue between the parties, including two terms contained in the '211 patent. The parties subsequently conducted discovery, and each party has filed two motions in the above-captioned 07-6053 case.

On August 3, 2009, Sun filed a Motion No. 2 For Summary Judgment Of Non-Infringement Of U.S. Patent No. 6,892,211 ("Motion No. 2") on the basis that its allegedly infringing product, Zettabyte File System ("ZFS"), does not practice either of the following two claim limitations present in every claim of the '211 patent as construed by this Court: (1) an on-disk root inode that "point[s] directly . . . to a first set of blocks on said storage system that store a first consistent state of said file system"; or (2) an incore root inode that "points directly . . . to . . . a second set of blocks on said storage system, . . . said second set of blocks storing data and metadata for a second consistent state of said file system." Motion No. 2 was fully briefed, and a hearing was held on September 23, 2009. Having considered the record in this case and the parties' statements at oral argument, and for the reasons set forth below, the Court hereby GRANTS Sun's Motion No. 2 For Summary Judgment Of Non-Infringement of the '211 patent.

II. LEGAL STANDARD

A. Summary Judgment

Summary judgment shall be granted if "the pleadings, discovery and disclosure materials on file, and any affidavits show that there is no genuine issue as to any material fact and that the movant is entitled to judgment as a matter of law." Fed. R. Civ. Pro. 56(c). Material facts are those which may affect the outcome of the case. See Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248 (1986). A dispute as to a material fact is genuine if there is sufficient evidence for a reasonable jury to return a verdict for the nonmoving party. Id. The court must view the facts in the light most favorable to the non-moving party and give it the benefit of all reasonable inferences to be drawn from those facts. Matsushita Elec. Indus. Co. v. Zenith Radio Corp., 475 U.S. 574, 587 (1986). The court must not weigh the evidence or determine the truth of the matter, but only determine whether there is a genuine issue for trial. Balint v. Carson City, 180 F.3d 1047, 1054 (9th Cir. 1999).

A party seeking summary judgment bears the initial burden of informing the court of the basis for its motion, and of identifying those portions of the pleadings and discovery responses that demonstrate the absence of a genuine issue of material fact. Celotex Corp. v. Catrett, 477 U.S. 317, 323 (1986). Where the moving party will have the burden of proof at trial, it must affirmatively demonstrate that no reasonable trier of fact could find other than for the moving party. On an issue

where the nonmoving party will bear the burden of proof at trial, the moving party can prevail merely by pointing out to the district court that there is an absence of evidence to support the nonmoving party's case. Id. If the moving party meets its initial burden, the opposing party "may not rely merely on allegations or denials in its own pleading;" rather, it must set forth "specific facts showing a genuine issue for trial." See Fed. R. Civ. P. 56(e)(2); Anderson, 477 U.S. at 250. If the nonmoving party fails to show that there is a genuine issue for trial, "the moving party is entitled to judgment as a matter of law." Celotex, 477 U.S. at 323.

B. Patent Infringement

"To prove infringement, the patentee must show that the accused device meets each claim limitation either literally or under the doctrine of equivalents." Catalina Mktg. Int'l v. Coolsavings.com, Inc., 289 F.3d 801, 812 (Fed. Cir. 2002). A determination of infringement, whether literal or under the doctrine of equivalents, is a question of fact. Id. "Literal infringement requires the patentee to prove that the accused device contains each limitation of the asserted claim." Id. "Summary judgment of no literal infringement is proper when, construing the facts in a manner most favorable to the nonmovant, no reasonable jury could find that the accused system meets every limitation recited in the properly construed claims." Id. Where the parties do not dispute any relevant facts regarding the accused product, but disagree over possible claim interpretations, the question of literal infringement collapses into claim construction and is amenable to summary judgment. General Mills, Inc. v. Hunt-Wesson, Inc., 103 F.3d 978, 983 (Fed. Cir. 1997); cf. Int'l Rectifier Corp. v. IXYS Corp., 361 F.3d 1363, 1375 (Fed. Cir. 2004) (distinguishing General Mills on the basis that only the structure of the accused devices had been stipulated to, not the disputed factual determination of whether the device met the claims as construed, but not addressing the scenario in which no reasonable juror could find that a certain claim limitation was met).

In MyMail Ltd. v. America Online, Inc., 476 F.3d 1372, 1378 (Fed. Cir. 2007), the Federal Circuit reviewed a District Court order granting summary judgment of non-infringement. Because there were no material factual disputes as to the operation of the accused systems, and the parties' disagreements concerned whether the defendants' systems performed "authentication" as defined by the patent and construed by the district court, the Federal Circuit found that the issue reduced to a

question of claim interpretation and affirmed summary judgment. See id. (noting that the accused product did not satisfy the authentication requirement as it did not validate the user's ID and password, as required by the patent's authentication process). These cases teach that the Court cannot leave it to the jury to decide the proper scope of the patent claim terms. 02 Micro Int'l Ltd. v. Beyond Innovation Tech. Co. Ltd., 521 F.3d 1351, 1360 (Fed. Cir. 2008) ("When the parties raise an actual dispute regarding the proper scope of the[] claims, the court, not the jury, must resolve the dispute.").

III. Sun Has Established That No Reasonable Trier Of Fact Could Find Literal Infringement of U.S. Patent No. 6,892,211

A. The Patent At Issue

The '211 patent is a continuation of the NetApp's 5,819,292 patent and is similarly directed to a method for keeping a file system in a consistent state. The '292 patent describes a method for maintaining consistent states of a file system, and for creating snapshots that are read-only copies of the file system. '292 Patent Abstract. The '211 patent contains more detailed mechanics of a file system that progresses from one "consistent state" to another. The system utilizes a root inode that contains pointers that directly point to a metadata file known as the inode file that contains the inodes of all of the other files in the file system. Homrig Decl., Ex. 2 ('211 Patent) at 9:25-33.

The '211 patent includes three independent claims – claims 1, 9, and 17 – each of which is asserted against Sun. NetApp also asserts dependent claims 2, 3, 10, 11, 18, and 19.

The three independent claims are similar, and Claim 9 is for:

> a device comprising:
> a processor;
> a memory; and
> a storage system including one or more hard disks;
> wherein said memory and said storage system store a file system; and
> wherein said memory also stores information including instructions executable by said processor to maintain said file system, the instructions including steps of (a) maintaining an on-disk root inode on said storage system, **said on-disk root inode pointing directly** and indirectly **to a first set of blocks on said storage system that store a first consistent state of said file system**, and (b) maintaining an incore root inode in said memory, **said incore root inode pointing directly** and indirectly to buffers in said memory and **a second set of blocks on said storage system,** said buffers and **said second set of blocks storing data and meta-data for a second consistent state of said file system**, said second set of blocks including at least some blocks in said first set of blocks, with changes between said first consistent state and said second consistent state being stored in said buffers and in ones of said second set of blocks not pointed to by said on-disk inode.

Id. at 24:39-62 (emphasis added).

The Court has construed the term "pointing directly and indirectly to buffers in said memory and a second set of blocks on said storage system" as "pointing directly and indirectly to buffers in said memory and pointing directly and indirectly to a second set of blocks on said storage system." 9/10/08 Order at 45, 50. The primary dispute between the parties was whether or not this term requires both direct and indirect pointing to both blocks and buffers, or can be satisfied by either indirect or direct pointing to either blocks or buffers. Id. at 45-46. The Court agreed with Sun that the patent requires both direct and indirect pointing to both blocks and buffers. Id. at 50.

B. NetApp's Infringement Contentions and Sun's Motion

Sun's accused ZFS product is a general use file system technology that is incorporated into Sun's open-source OpenSolaris operating system and its counterpart, Solaris 10 OS. ZFS operates as a storage pool that supports both file systems and other types of datasets. Multiple file systems can exist simultaneously within a common storage pool managed by ZFS. McKusick Decl. ¶ 6-7[1]; Williamson Decl., Ex. 2 (ZFS On-Disk Specification) at 5.

According to NetApp, the "uberblock" is the on-disk root inode of the '211 patent. It is undisputed that the uberblock has only one active pointer, which points directly to and only to one part of the "meta object set" ("MOS") of the storage pool. Williamson Decl., Ex. 4 (Bonwick Depo.) at 226-27; Ex. 2 at 31-32 ("The DSL is implemented as an object set of the type DMU_OST_META. This object set is often called the Meta Object Set, or MOS. There is only one MOS per pool and the uberblock (see Chapter One) points to it directly."). More specifically, the uberblock's pointer points to a block at the top of the MOS that contains an objset_phys_t structure of the type DMU_OST_META (referred to herein as the "objset meta structure"), which is part of the MOS. The MOS describes and manages relationships between and properties of object sets. Id. at 29.

ZFS creates four kinds of object sets: file systems, clones, snapshots, and volumes. Id.

---

[1]Dr. McKusick's Declaration was submitted in support of Sun's Motion No. 1 For Summary Judgment of Non-Infringement of U.S. Patent No. 6,892,211. However, to the extent that it is relevant and helpful to explain the background of the technology at issue in Motion No. 2 as well, the Court has relied upon it.

These four types of object sets reside below the MOS, which manages their relationships. Id. at 29-30. Numerous objects sets can reside below the MOS. They are used in ZFS to group related objects, such as objects in a filesystem, snapshot, clone, or volume. Id. at 26.

NetApp's position is that the claim limitation that requires the "on-disk root inode" to point directly to blocks that store a first consistent state of said file system is practiced in ZFS by the uberblock pointing directly to the "objset_phys_t data structure" for the MOS. Williamson Decl., Ex. 3 (Ganger Report) at ¶¶ 46, 83, 122. NetApp further asserts that the ZFS spa_t structure is the claimed "incore root inode" of the patent, and that the requirement that such an inode point directly to blocks storing data and metadata for a second consistent state of said file system is practiced by the spa_t structure pointing to the objset_phys_t structure in the MOS. Id. at ¶¶ 49, 86, 125[2]; see also NetApp Opp. at 2. NetApp contends that the uberblock and spa_t point directly to the objset_phys_t data structure for the MOS and indirectly to other blocks in the file system. The spa_t also points directly and indirectly to buffers in memory. Homrig Decl., Ex. 32 (Ganger Report) at ¶¶ 46, 49-50.

Sun counters that, based on the Court's construction, any alleged on-disk or incore root inode must point directly to blocks that store at least part of a consistent state of said file system. Sun argues that the structure on Sun's ZFS software that NetApp asserts is the on-disk root inode (the uberblock) does not point "directly" to blocks that store a first consistent state of the file system, and the structure it contends is the incore root inode (the spa_t structure) does not point "directly" to blocks storing data and metadata for a second consistent state of the file system. According to Sun, because the claims require direct pointing to blocks that store a consistent state of a file system, and because the only direct pointing is pointing to objset_phys_t in the MOS, NetApp necessarily contends that objset_phys_t (objset meta structure) in the MOS is part of a ZFS file system, which it is not.

The following facts are undisputed:

1) NetApp contends that the ZFS uberblock is the claimed on-disk root inode;

---

[2] Prior to claim construction, NetApp originally contended that a dnode at the bottom of the MOS was the on-disk root inode, specifically the dnode_phys_t of the type DMU_OT_DLS_DATASET. Williamson Decl., Ex. 7 (NetApp Infringement Contentions) at 3, 4, 17.

2) NetApp contends that the ZFS spa_t structure is the claimed incore root inode;

3) The only structure in ZFS to which the uberblock and the spa_t structure point directly is the objset_phys_t structure of the type DMU-OST-META ("objset meta" structure);

4) The objset meta structure is part of the ZFS Meta Object Set ("MOS"); and

5) The parties do not dispute the actual structure of ZFS insofar as it is relevant to this motion (i.e., the components and their arrangement) as described above and in Sun's Motion No. 2 at pages 7 through 11.

The parties also agree that the only issue for the Court to decide is whether the objset meta structure (which is part of the MOS) is part of a "file system," as claimed in the '211 patent. NetApp Opp. at 3:6-7; Sun Reply at 5:9-11. Sun argues that this issue is a question of law for the Court to decide on summary judgment; NetApp contends that if reasonable minds could differ then it is a fact question for the jury. The Court can decide this issue as a matter of law, and need not weigh conflicting evidence on which reasonable minds might disagree. The parties are in agreement as to the relevant components and structure of the ZFS system (though they disagree as to the proper labels for those components), and the resolution of this motion requires only a comparison of the properly construed claims to the accused technology. See General Mills, Inc. v. Hunt-Wesson, Inc., 103 F.3d 978, 981-83 (Fed. Cir. 1997).

C. Sun Has Met Its Initial Burden Of Presenting Evidence That The MOS Is Not A File System

In its opening brief, Sun argues that the MOS itself is not part of any ZFS file system, as its role is to manage a storage pool containing the different object sets available in ZFS, such as file systems, clones, snapshots, and volumes. According to Sun, file systems and ZFS filesystems, on the other hand, are well-defined object sets, distinct from poolwide data, that reside below the MOS, as described by the ZFS Specification. See Williamson Decl., Ex. 2 at 29 (ZFS filesystem is one of four types of object sets); 45 (describing ZFS filesystems); NetApp Opp. at 2 ("The MOS, which NetApp contends is part of a file system, manages the metadata and data comprising the tree of blocks contained with ZFS.") (citing ZFS Specification at 29). Sun also proffers testimonial evidence of ZFS developers indicating that the ZFS file systems are only one part of ZFS technology,

and that they are object sets that reside below the MOS. See, e.g., Williamson Decl., Ex. 6 (Maybee Depo.) at 164-66 (the MOS has data sets that point to file systems); 176, 186-92 (ZFS is generic term for technology but also used to label individual file systems within the pool, describing file systems below the MOS and that file system is pointed out by the MOS); 208-09. Sun also cites Mr. Bonwick's testimony in support of its argument that file systems are object sets separate from the MOS. See id. Ex. 5 (Bonwick Depo.) at 45-47 (file system means POSIX[3] layer but can be used to mean any object set or data set), 67 (file system means POSIX layer file system and the more generic data structure in ZFS is data set of the object set), 81, 108-13 (ZFS presents a pool of storage that many file systems can use; ZFS is a package that gives ability to create storage pools and file systems within the pool and zvols within the pool); Ex. 4 (Bonwick Depo.) at 228-29 (there are typically many active file systems in ZFS at any one time).[4]

NetApp challenges Sun's initial showing by asserting that Sun's own product specifications and related developer testimony about the nomenclature of its product are suspect because they might have been prepared with legal arguments in mind. Since Sun has cited no other evidence, according to NetApp, Sun has not met its initial burden. NetApp Opp. at 4. However, NetApp has put forth no evidence to support this speculation. Additionally, NetApp's own expert Dr. Ganger

---

[3] POSIX is an acronym for Portable Operating System Interface for UNIX. It is an IEEE standard that defines a set of operating system services. See Microsoft Computer Dictionary 351 (Fourth Ed. 1999).

[4] Sun also argues that NetApp's position regarding claim 8 of the '292 patent undercuts its claim and supports Sun's position because, there, Dr. Ganger refers to a plurality of read only copies of the file system, which means the POSIX file system (i.e., the below-the-MOS self-contained file system). See Williamson Decl., Exh. 3 (Ganger Report) ¶ 169 (ZFS supports creation of a plurality of snapshots (read-only copies) of a POSIX file system). Sun argues that this portion of Dr. Ganger's testimony correctly identifies ZFS file systems as being object sets residing below the MOS. NetApp takes an inconsistent position with respect to the '292 patent when it contends that the ZFS file system is rooted by an object that includes a dsl_dataset_phys_t data structure. Id. at ¶ 175. This object, according to Sun, is the denode_phys_t of the type DMU_OT_DSL_DATASET that resides at the bottom of the MOS. This object, in turn, points to file system object sets that reside below the MOS. Specifically, Dr. Ganger opines that each consistent state in ZFS is "rooted by an object that includes a dsl_dataset_phys_t data structure in the bonus buffer of its dnode. The dsl_dataset_phys_t data structure contains metadata for each successive state, as construed by the Court." At oral argument, NetApp conceded that Dr. Ganger was referring to the POSIX file system and agreed that the MOS is not copied, but argued that its position is not inconsistent because even though the POSIX file system is one file system, other things can constitute file systems as well. While these two NetApp positions are somewhat inconsistent, any inconsistency would go to Dr. Ganger's credibility, which the Court cannot assess in a motion for summary judgment.

relied heavily on the ZFS specifications in his analysis, as do NetApp's opposition papers. The Court therefore will not disregard Sun's initial evidentiary showing, and concludes that Sun has met its initial burden of presenting evidence that the MOS is not part of a file system.

D. The ZFS Specifications and Other "Labels" Do Not Raise A Triable Issue Of Fact That The MOS is Part of a File System

Though NetApp challenges Sun's use of product specification and "labels" to define the functions of the device, NetApp also counters Sun with other labels. NetApp argues that ZFS has been described as a file system in its entirety, as its full name Zettabyte File System bears out. See e.g., Homrig Decl., Ex. 49 (Solaris ZFS Administration Guide January 2009) at 41-42 (describing ZFS as most scalable file system ever); Ex. 20 (Bonwick Zettabyte File System Article) at 1, 3, 10 (describing ZFS as a file system); Ex. 32 (Ganger Report) at ¶¶ 23-25 ("ZFS is a general-purpose file system"; "ZFS is the native file system"), 32-33 ("ZFS . . . operates as a file system"; noting that Sun touts ZFS as a new file system). NetApp points to Dr. Ganger's testimony that Sun's own literature refers to ZFS as a whole as a file system, and notes that a file system may be divided into smaller file systems, as a set can contain sets that are subsets of it. Id. ¶ 33. NetApp's logic appears to be that, if ZFS as a whole has been labeled a file system, and the MOS is a part of ZFS, then the MOS is necessarily part of a file system as well.

However, NetApp has put forth no evidence specifically defining the MOS as a file system (other than as part of the whole ZFS product). And it is undisputed that ZFS does more than simply provide a file system – it operates as a pool of storage that supports both file systems and volumes, and manages the relationships between file systems. See Williamson Decl., Ex. 2 at 5, 29. If ZFS does more than merely provide a file system, it necessarily follows that some portion of the device is not part of a file system. Additionally, NetApp's expert Dr. Ganger agreed with the proposition – as does the Court – that, "a system can be called a file system by its authors, but that doesn't necessarily make it a file system for purposes of the ... '211 patent[]." Williamson Reply Decl., Ex. B (Ganger Depo) at 170. Sun also persuasively argues that the fact that certain marketing materials that refer to a "file system" or that ZFS was referred to as a file system as a whole cannot preclude summary judgment, because the file systems that reside below the MOS constitute a key feature of ZFS,

whereas it is commonplace for marketing materials and nomenclature to simplify complex technical products. See Williamson Decl., Ex. 5 (Bonwick Depo) at 111 (noting that "file system is a technology people are familiar with. That is one of the capabilities that ZFS provides. So it seemed a reasonable name."). In other words, a mere label in itself, especially one geared towards customers, does not constitute evidence of the MOS' actual status (or lack thereof) as a file system under the meaning of the patent sufficient to raise a triable issue of fact.

NetApp also argues that Sun's own witnesses concede that ZFS is a file system. NetApp points to statements made by Matthew Ahrens, an early and central member of the ZFS design team, that the "entire filesystem is represented on disk as a giant tree of blocks." Id., Ex. 38 at 2. However, this testimony is not specifically directed to the MOS and does not state what is or is not part of the file system and is therefore not helpful to NetApp. NetApp also notes that Jeff Bonwick, chief architect on the ZFS team, confirmed that the ub_rootbp pointer contained in the uberblock is the root of the tree of blocks. Id., Ex. 16 at 22 (noting that pointer in uberblock is the root that points to tree of blocks in storage system, and that what is "actually being pointed at" is the "objset phys_t for the meta object set of the storage pool"). NetApp argument appears to be that, because the parties agree that the pointer in the uberblock points directly to the MOS, and because the above testimony states that the uberblock pointer points to a tree of blocks, one can infer that the MOS is itself part of the tree of blocks that comprises the ZFS file system. The logic of this argument is tenuous at best, and also does not create a triable issue of fact.

E. Any Alleged "Disagreement" Between The Experts Does Not Preclude Summary Judgment

NetApp also argues that the testimony of Sun's own expert, Dr. McKusick, is sufficient to defeat Sun's motion. NetApp contends that the parties' experts disagree on whether the MOS is part of the file system, and that Dr. McKusick conceded that reasonable minds could disagree about the factual question of whether the MOS is part of a file system. NetApp Opp. at 5. NetApp relies on Dr. McKusick's testimony that he respects Professor Ganger and he is knowledgeable about ZFS code, and then argues that, "[a]fter analyzing the evidence, Professor Ganger reached the conclusion that the MOS is part of a file system." Id.

This argument fails for two reasons. First, Dr. Ganger did not actually testify that the MOS is part of a file system. Rather, he opined that "ZFS as a whole is a file system rooted by an on-disk uberblock." Ganger Report ¶ 43 (also noting that data structures in ZFS refer to dnodes in the MOS and that the uberblock points to the MOS). Second, Dr. McKusick disagreed with this conclusion, explaining that their opinions differed because they disagreed about the functionality or characterization of the code making up ZFS – specifically as to whether or not certain functionality was part of the MOS or part of the ZFS files management. Homrig Decl., Ex. 1 (McKusick Depo.) at 31. NetApp also paraphrases Dr. McKusick's as acknowledging that "reasonable minds may disagree about the factual question of whether the MOS is part of a file system," but this somewhat distorts his testimony. Rather, Dr. McKusick stated in very general terms that the experts analyzed the evidence and came to different conclusions, and testified politely that reasonable minds could differ on issues where they had disagreements as reflected in the expert reports. Id. at 36-37. However, Dr. McKusick further testified that where he did not agree with Dr. Ganger's opinions, he could not characterize them as reasonable or unreasonable, but only stated that he did not agree with him and that his opinions were incorrect. Id. at 32-33. More importantly, nowhere in the testimony cited by NetApp was Dr. McKusick asked about file systems, whether the MOS is part of the file system, whether Dr. Ganger's opinions about the MOS being part of the file system are reasonable, or whether reasonable minds could differ on this particular question. Id. Instead, Dr. McKusick testified that Dr. Ganger's conclusions were incorrect (id. at 33-34), and pointed out that the experts do not dispute how the ZFS software code works – the facts – but rather dispute one another's characterizations of the functionality (id. at 31).[5]

NetApp then relies on Amini Innovation Corp. v. Anthony California, Inc., 439 F.3d 1365, 1368 (Fed. Cir. 2006) for the proposition that summary judgment is improper where reasonable minds could differ. NetApp Opp. at 1. However, Amini is a copyright case, not a patent case, that held on a different legal issue that the "protection of a particular design under copyright law raises a

[5]Courts must be attuned to the strategic incentives that shape expert testimony. If the mere expression of this type of general respect for an opposing expert in high stakes litigation, where both sides can afford to hire well-credentialed experts, were deemed sufficient to defeat summary judgment, there would be undue pressure on dueling experts to engage in disrespectful, diversionary hyperbole.

mixed question of law and fact" and that "[w]here reasonable minds could differ on the issue of substantial similarity, however, summary judgment is improper." Id. at 1368 (emphasis added). Because the parties here agree on the actual structure of the relevant components of ZFS and how they operate, the question is one of claim scope and can be decided on summary judgment. NetApp's argument on this point is not grounds for denying summary judgment.

F. Summary Judgment Is Proper Because The Evidence Shows That The MOS Is Part of a File System Under Either Party's Proposed Construction of "File System"

The Court may grant summary judgment of no literal infringement if, construing the facts in the manner most favorable to the nonmovant, no reasonable jury could find that the accused system meets every limitation of the claims. See, e.g., Catalina Mktg. Int'l v. Coolsavings.com, Inc., 289 F.3d 801, 812 (Fed. Cir. 2002). As discussed below, based on the evidence presented, no reasonable factfinder could find that the "file system" limitation of the '211 patent is met when comparing that claim term, whether construed as proposed by Sun or as proposed by NetApp, with the allegedly infringing ZFS device.

1. Sun's proposed construction: "a collection of files and file management structures"

The Court has not yet construed the term "file system," for which the parties propose different constructions. Nonetheless, the parties agree that it is unnecessary for the Court to construe that term now, because under either construction each party contends that it should prevail. Sun has proposed that the term "file system" should mean "a collection of files and file management structures." NetApp argues that the MOS meets this definition because ZFS "objects" are "files," and "object sets" residing below the MOS are represented by files, and the MOS is a "file management structure" that manages these "files." NetApp contends that Sun's expert has defined "volumes" as essentially large files, and the other three types of object sets below the MOS (filesystems, clones and snapshops) as file systems. According to NetApp, the MOS manages these object sets (which it argues have been defined by Sun's expert as files and file systems) through functions such as an "object directory" (which maps from an object number to an object below the MOS) and a "dataset directory" or a DSL directory (which manages related groupings of datasets and properties associated therewith). NetApp's position is that the MOS is a "collection of files and file

management structures" because it manages at least one type of file (volumes) directly, and other types (filesystems, clones and snapshots) through management of those object sets.

NetApp's argument fails for several reasons. First, it relies on testimony that does not really support its statements. NetApp characterizes Dr. Ganger's testimony as stating that ZFS objects are files and object sets are represented by files in ZFS. However, NetApp cites sections of Dr. Ganger's report that do not support its argument in that they are not specifically directed to the MOS, are conclusory, or do not clearly explain his position about what constitutes a file. See, e.g. Homrig Decl., Ex. 32 (Ganger Expert Report) ¶ 184 (conclusory statement referring to "object (flat file)" without explanation in context of '292 patent), ¶¶ 23-25 (referring generally to ZFS as a file system and its use of a flexible file system organization, but not referring to the MOS specifically), ¶¶ 32, 43 (noting that Sun's literature refers to ZFS as a whole as a file system, and that a file system may be subdivided into smaller file systems; further noting that ZFS refers to roots of its file systems (dnodes in the MOS)); ¶ 49 (stating generally that field in spa_t uberblock structure points to blocks of storage); see also Ex. 20 (Bonwick article) (generally equating "objects" with "flat files" without explanation).

To the extent that NetApp's arguments are based on Dr. Ganger's expert report, Dr. Ganger never actually opines that the MOS is part of the claimed file system. Dr. Ganger did testify at his deposition that "the file containing the objset_phsy_t is a file," but there he gave no opinion as to why he believed the objset_phsy_t is in a file, what file it is in, or that the objset_phys_t is itself a file. See Homrig Decl. Ex. 33 at 690. Moreover, the objset_phys_t is only one of numerous structures in the MOS. Finally, Dr. Ganger's own testimony that one indicia of a "file system" is that it creates, reads, writes, and deletes files (and that those functions are performed by a user of the file system) (Williamson Reply Decl., Ex. B at 252-53) points against NetApp's arguments, as Dr. Ganger admitted that the MOS does not store user files that a user can read, modify, or write. Id. at 256-57.

NetApp also asserts that Dr. McKusick concedes that the MOS manages at least one type of file (ZFS volumes) directly, because he allegedly admitted that ZFS volumes are "essentially large files." NetApp. Opp. at 8 (paraphrasing Dr. McKusick's testimony located at Homrig Decl., Ex. 1 at

191-92). NetApp further argues that Dr. McKusick considers the other three types of object sets below the MOS (ZFS file system, clone, and snapshot object sets or file systems) to be file systems. Id. at 189-92, 207. However, NetApp's characterization of Dr. McKusick's statements is also not entirely accurate. He did not state that ZFS volumes are "essentially large files" and that the other types of object sets are file systems. Rather, he stated that ZFS Volume is not a "file system" and that it looks more like a file than a file system, but he does not actually state that it is either a file or a file system. Upon further questioning, he states that a volume is an object set that "can be" treated as a large file and that the MOS doesn't manage it. Id. at 193-94.

Additionally, all of these object sets (volumes, clones, snapshots and filesystems) reside below the MOS, so even if NetApp's characterization of Dr. McKusick's testimony was accurate, it does not advance NetApp's argument. Sun also notes that this evidence shows that the MOS does not manage ZFS volumes directly. See Williamson Reply Decl., Ex. A (McKusick Depo) at 193-94. Dr. McKusick, when asked if the MOS manages objects, stated, "It doesn't manage the Z vol any more than it manages the file systems. It has a pointer to it, it provides blocks to it, but the management of the blocks and the structuring of the blocks is all managed down below that line which represents the MOS." Id. at 194. In sum, the evidence cited by NetApp discussed above does not create a triable issue of fact.

Second, NetApp has not shown that either the "object directory" or the "DSL directory" functions of the MOS constitute "file management." Specifically, NetApp argues that the MOS contains a structure that carries out file management functions called the "object directory." Williamson Decl., Ex. 2 (ZFS On-Disk Specification) at 31 (noting that there is a single distinguished object in the meta object set, called the object directory). The object directory contains a file management structure called a Dataset and Snapshot Layer (DSL) Directory. Id. at 29-31. These directories "manage a related grouping of datasets and the properties associated with that grouping." Id. at 30. NetApp contends that Dr. McKusick conceded that these directories directly control files by conceding that the MOS controls whether files can be executed on a data set. See id., Ex. 1 (McKusick Depo.) at 205. According to NetApp, the MOS also contains information about location, date of creation, and pointers containing information about portions of ZFS that Dr.

McKusick acknowledges are file systems. Id. at 163-168.

Sun counters that the object directory and DSL directory relied on by NetApp as structures that meet the "file management structure" part of this construction are not related to management of files in a file system. The ZFS specification does identify ways in which the DSL directory within the MOS controls datasets, but not individual files. See id., Ex. 2 at 35-36 (noting, for example, that the directory "controls inheritance behavior for datasets," "controls whether files can be executed on a dataset," and "controls whether objects can be modified on a dataset"). Sun argues that the object directory simply contains a number that corresponds to a number given to an object set that allows the object set to be located. NetApp Opp. at 9 (citing Ex. 1 McKusick Depo at 151-52 ("object directory takes, it maps from an object number to a – an object that's in the MOS"). Thus, the object directory is unrelated to management of files in a file system, according to Sun. A dataset directory, as noted by the ZFS specification and NetApp, manages a grouping of datasets. ZFS Specification at 30; NetApp Opp. at 9. Each dataset is associated with a specific object set, and the MOS manages groups of object sets, and is not part of a file system managing the files within that system. ZFS Specification at 35; Sun Reply at 12.

Sun's position is correct. The dispute between the parties on this issue is whether a "file management structure" must be within a file system managing files, or can be a higher level management function between file systems. The plain language of the claim (and Sun's proposed construction thereof) contemplates management of files within a file system, not the higher level organization or grouping of various object sets or file systems. The MOS structures that perform management functions concerning object sets manage the inter-relationships between object sets (file systems), identify or maintain statistics about the object sets below the MOS, or impose properties on a group of object sets. But these structures do not manage files within an individual object set, which is left to that individual object set, or file system. See Williamson Reply Decl., Ex. A (McKusick Depo) at 207-08, 216-17 (individual ZFS file systems manage files in the file systems); ZFS Specification at 45 ("POSIX is a standard defining the set of services a file system must provide. ZFS file systems provide all of these required services.").

In sum, NetApp has not raised a triable issue of fact that the MOS constitutes a "collection of

files and file management structures" to defeat summary judgment.

2. NetApp's proposed construction: "a system that logically organizes file-related information (such as directories, files and blocks) in a tree structure"

With respect to NetApp's proposed construction of the term "file system" as "a system that logically organizes file-related information (such as directories, files and blocks) in a tree structure," the Court first notes that the language "file-related information," even with the parenthetical, likely will be too vague and overbroad for the Court to adopt if further construction is undertaken. However, the Court has considered whether a reasonable jury could find that the allegedly infringing device meets this claim as construed by NetApp.

NetApp argues that ZFS is a giant tree of blocks and the MOS is part of this "tree-structure." The parties do not dispute this point. NetApp relies on Dr. Ganger's report and deposition for the additional point that the MOS itself is represented by a file. See Homrig Decl., Ex. 32 (Ganger Report) ¶ 184, 23-25, 32-46; Ex. 33 (Ganger Depo.) at 688 (testifying that the objset_phys_t_data structure for the MOS is stored in a file). However, as discussed in the previous section, Dr. Ganger's testimony does not fully support NetApp's position.

NetApp further contends that the MOS contains at least two types of directories (the object directory and DSL directory) and that the information contained therein is "file-related." NetApp relies on Dr. McKusick's testimony to summarily assert that "the directories within the MOS are file-related." Opp. at 11 (citing McKusick Depo. at 205). However, in this portion of his deposition, Dr. McKusick states that the MOS "controls whether files can be executed on a data set," but he later clarifies (with respect to another function of the MOS) that the MOS is "not managing the files in the filesystem. This is a directive to the file system on how it should manage its files." Id. at 207. Additionally, and as discussed in the previous section, where NetApp relies on Dr. McKusick's testimony to assert that all objects are files, the testimony cited does not support this assertion.

As discussed above with respect to Sun's definition, the evidence shows that the MOS manages object sets and how they operate and interact with each other, but not individual files. NetApp attempts to blur this important distinction between organizing file-related information (such as directories, files and blocks), and file system-related information (the higher level management

that the MOS performs). NetApp has not shown that a reasonable jury could conclude that the claim requirement is met, even under its own proposed construction, and summary judgment is appropriate.

3. The Objset Meta Structure Has Not Been Shown To Be A File System

In its Reply papers, Sun argues that NetApp has also not shown that the objset meta structure specifically (i.e., what the uberblock points directly to at the top of the MOS), as opposed to the MOS generally, qualifies as a file system structure. It is undisputed that the objset meta structure is a 1 KB structure stored in a block. Williamson Decl, Ex. 2 (ZFS Specification) at 26, 32. There is no evidence that this structure is a file or a collection of files or that it manages any files. NetApp's only evidence supporting its position that this particular structure is a file system is Dr. Ganger's statement that the objset_phys_t data structure for the MOS is stored in a file. NetApp Opp. at 10:24-25 (citing Homrig Decl., Ex. 33 at 688-89) ("my opinion that the – the file containing the objset_phys_t is a file"). However, Dr. Ganger did not state that the objset meta structure itself is a file, manages files, or organizes file-related information. Nor does this testimony establish that it is file-related under NetApp's proposed construction, because Dr. Ganger does not opine that something stored in a file is necessarily file-related. In addition, his assertion that the structure is stored in a file is conclusory, as he has not identified the file, explained the assertion, or pointed to evidence in support of this conclusion. He merely stated, "because that's how I understand ZFS to work. It uses files to store things like the objset_phys_t, for example." This is insufficient to create a triable issue of fact. Sun also notes that there is no evidence showing that the objset meta structure is even an object, which it is not, because it is a separate structure that describes an object set. Williamson Decl., Ex. 2 at 26 § 3.2 (further stating that objsets are represented by a 1 kilobyte object_phys_t structure). Therefore, to the extent that the MOS generally could be considered a file system, which as discussed above it cannot, summary judgment is still appropriate on the basis that the objset meta structure has not been shown to be a file system under either parties' construction.

4. Sun's Argument that A File System Cannot Be Itself And Also a Portion Of Itself

On reply, Sun also notes that the '211 patent teaches that the MOS is not part of the claimed

file system. The patent claims a single, specific file system as the invention. See, e.g., Homrig Decl., Ex. 2 ('211 Patent) at 4:11-12 ("present invention provides a method for maintaining a file system . . ."); 20:15 ("the file system of the present invention"). Sun argues that NetApp's position is that the claimed "file system" is both itself (the file system) and a portion or subset of that same file system (the MOS), which is illogical and nonsensical. Sun points out that NetApp only cites one example of a file system which can both be itself and part of itself; in footnote 4 of its Opposition. There, NetApp points to prior art cited in the '211 patent (the Episode file system), but Sun correctly asserts that this prior art is not the claimed invention, and that in any event the prior art refers to a "plurality of separate" file systems, as opposed to a discussion of a first file system and second file system that is a subset of that first system as NetApp argues here. Id. at 1:47-48. It is also not clear from the patent that the Episode file system as a whole is referred to as a file system, except to the extent that, like ZFS, the name of the technology as a whole contains the phrase "file system" within it, which is not persuasive for the reasons given above. Id. at 1:40-65.

NetApp countered at the hearing that a file system may be part of a file system just as Dr. Ganger testified that a subset may be part of a set. See Williamson Decl., Ex. 3 ¶ 33. However, this analogy does not help NetApp. Under basic set theory, which can be taught before high school (e.g., with a Venn diagram), these terms describe a relationship in which a proper subset is contained within a larger set. See Wikipedia The Free Encyclopedia, http://en.wikipedia.org/wiki/Set_theory (last visited Nov. 11, 2009); http://en.wikipedia.org/wiki/Subset (last visited Nov. 11, 2009). But consistent with set theory, the larger set here differs from and includes something more that the subset it contains. As explained above, here it is undisputed that ZFS as a whole does more than the file systems that it includes below the MOS, such as manage the relationships between the file systems, and also manage other components besides file systems such as volumes. Thus, this argument lends further support to the Court's conclusion that summary judgment should be granted.

//

//

//

//

For all of the foregoing reasons, Sun's Motion No. 2 For Summary Judgment Of Non-Infringement of the '211 patent is hereby GRANTED.

IT IS SO ORDERED.

Dated: November16, 2009

ELIZABETH D. LAPORTE
United States Magistrate Judge

UNITED STATES DISTRICT COURT

FOR THE

NORTHERN DISTRICT OF CALIFORNIA

NETWORK APPLIANCE INC,

Plaintiff,

v.

SUN MICROSYSTEMS INC et al,

Defendant.
_____/

Case Number: CV07-06053 EDL

CERTIFICATE OF SERVICE

I, the undersigned, hereby certify that I am an employee in the Office of the Clerk, U.S. District Court, Northern District of California.

That on November 17, 2009, I SERVED a true and correct copy(ies) of the attached via email to:

Counsel for Sun Microsystems: Christine Kerba Corbett
DLA Piper US LLP
christine.corbett@dlapiper.com

Counsel for Network Appliance: Edward Robert Reines
Weil Gotshal & Manges LLP
Edward.Reines@weil.com

Dated: November 17, 2009

Richard W. Wieking, Clerk

By: Lili M. Harrell, Deputy Clerk